

**Systém pro správu síťové
infrastruktury**
Network Management System

Zadání diplomové práce

Student: **Bc. Martin Patyk**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **System pro správu síťové infrastruktury
Network Management System**

Zásady pro vypracování:

Cílem této diplomové práce je návrh a implementace systému pro centrální správu síťové infrastruktury. Tento systém bude vytvořen na základě požadavků firmy zabývající se poskytováním telekomunikačních služeb.

Úkolem diplomanta bude implementovat informační systém, který se bude skládat ze dvou hlavních částí: První část bude tvořit webová služba, jejímž úkolem bude komunikovat se síťovými prvky, načítat z nich data a případně do těchto síťových prvků posílat příkazy. Komunikaci mezi síťovými prvky a webovou službou bude zajištěn protokolem SNMP nebo Mikrotik API.

Druhá část práce se bude zabývat návrhem a implementací webové aplikace, která bude zprostředkovávat komunikaci mezi webovou službou a uživatelem, a zároveň vizualizovat data získané webovou službou ze síťových prvků.

Teoretická část práce bude obsahovat přehled stávajících řešení pro správu síťové infrastruktury, souhrn výhod i nevýhod stávajících řešení a hlavně analýzu a popis vlastní implementace.

Předpokládá se splnění následujících bodů:

1. Popis a porovnání stávajících řešení na trhu.
2. Soupis jejich výhod a nedostatků.
3. Vývoj/analýza komponenty pro komunikaci s routery (návrh rozhraní).
4. Nástroj pro zobrazení a správu topologie sítě (graf připojených routerů a PC).
5. Návrh rozhraní pro úpravu rychlosti linky.
6. Návrh a implementaci tvorby grafů přenesených dat na klienta.
7. Návrh a implementace komponent pro monitorování síťových komponent.
8. Návrh a implementace webového rozhraní pro správu systému.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Tomáš Kocyan**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2012

.....Martin Pásl.....

Rád bych na tomto místě poděkoval vedoucímu práce Tomáši Kocyanovi za podnětné připomínky a technickou podporu při tvorbě této diplomové práce. Také bych rád poděkoval své rodině za podporu a pomoc při její tvorbě.

Abstrakt

Diplomová práce se zabývá analýzou, návrhem a implementací aplikačního rámce sloužícího ke vzdálené správě aktivních síťových prvků. Analýza se zaměřila na zmapování výhod a nedostatků dosavadních řešení vzdálené správy síťových prvků. Na základě popsaných vlastností stávajících řešení vznikl návrh na vylepšení způsobu správy síťových prvků. Ohled byl především brán na budoucí rozšíření aplikace a oddělení prezentační části aplikace od části určené ke komunikaci se síťovými prvky. Implementační část je zaměřená na tvorbu webové služby, která nabízí prostřednictvím vzdáleného volání procedur možnost spravovat aktivní síťové prvky. K ověření funkčnosti webové služby byl vytvořen webový informační systém demonstrující práci s webovou službou a správu evidovaných síťových prvků.

Klíčová slova: Webová služba, Mikrotik API, ASP.NET, RRDtool, cron, PHP5, RPC

Abstract

This thesis deals with the analysis, design and implementation of application framework pertaining to the remote management of active network elements. The analysis focused on mapping the advantages and shortcomings of the current solution to remote management of network elements. Based on the described characteristics of existing solutions, it is proposed to improve the management of network elements. Consideration was mainly given to the future expansion of the application and presentation of the separation of applications designed to communicate with network elements. The implementation part is focused on creating a Web service, which offers via RPC to manage active network elements. To verify the functionality of web services was created web-based information system that demonstrates working with and managing Web service registered network elements.

Keywords: Web services, Mikrotik API, ASP.NET, RRDtool, cron, PHP5, RPC

Seznam použitých zkratk a symbolů

API	– Application Programming Interface
HTTP	– Hypertext Transfer Protocol
ICMP	– Internet Control Message Protocol
IP	– Internet Protocol
NNTP	– Network News Transfer Protocol
POP3	– Post Office Protocol
QoS	– Quality of Service
RPC	– Remote procedure call
RRDtool	– Round Robin Database Tool
SMTP	– Simple Mail Transfer Protocol
SNMP	– Simple Network Management Protocol
SSH	– Secure Shell
TCP	– Transmission Control Protocol
UDP	– User Datagram Protokol

Obsah

1	Úvod	9
1.1	Struktura diplomové práce	10
2	Přehled aplikací monitorujících síťové prvky	11
2.1	Nagios	11
2.2	The Dude	12
2.3	ISPadmin	13
3	Analýza a popis vlastního řešení správy síťové infrastruktury	17
3.1	Nové funkce stávajícího informačního systému	17
3.2	Charakteristika navrhovaného řešení	17
3.3	Oddělení sběru dat od jejich prezentace	19
3.4	Výhody navrhovaného řešení	19
4	Komponenta pro komunikaci se směrovači	21
4.1	Popis komponenty	21
4.2	Přihlášení ke směrovači prostřednictvím Mikrotik API	22
4.3	Popis třídy Mk	24
4.4	Analýza komponenty	24
4.5	Posílání příkazů směrovači	24
4.6	Obecná implementace	26
5	Rozhraní pro úpravu rychlosti linky	29
5.1	Traffic shaping	29
5.2	Traffic Policing	30
5.3	Implementace tvarování provozu	30
5.4	Řízení provozu na směrovači Mikrotik	31
5.5	Implementace - webové rozhraní	32
5.6	Návrh ovládaní rozhraní pro řízení rychlosti linky	32
5.7	Implementace úpravy rychlosti linky - webová služba	32
6	Tvorba grafů přenesených dat	35
6.1	Tvorba grafů	35
6.2	Popis programu RRDtool	35
6.3	Použití programu rrdtool	36
6.4	Vytvoření grafu	37
6.5	Návrh aplikace ke tvorbě grafu	38
6.6	Implementace skriptu	38
6.7	Souhrn použitého software a jeho verze	40

7	Monitorování síťových komponent	41
7.1	Stav směrovače	41
7.2	Stav síťových rozhraní	41
7.3	Návrh implementace monitorování síťových komponent	42
8	Návrh webového rozhraní pro správu systému	45
8.1	Využití webového rozhraní	45
8.2	Vstupy do systému	46
8.3	Výstupy ze systému	46
8.4	Nefunkční požadavky	46
8.5	Datová analýza	47
8.6	Implementace webového rozhraní	47
8.7	Použitý software k implementaci	48
9	Nástroj pro zobrazení a správu topologie sítě	49
9.1	Návrh evidence síťových prvků	49
9.2	Návrh nastavení a změna toku dat v síti	50
10	Závěr	51
11	Reference	53

Seznam tabulek

1	Tabulka operačních systémů běžících na použitých směrovačích	18
2	Verze Ethernetu a její rychlost	29
3	Typ hodnot proměnných v databázi RRDtool	36
4	Základní parametry tvorby grafu	37
5	Typ a verze použitých programů	40
6	Datový slovník tabulky mikrotik	47
7	Typ a verze použitých programů	47

Seznam obrázků

1	Ukázka aplikace The Dude	12
2	Náhled na návrh aplikace	17
3	Přístup z různých druhů zařízení	18
4	Blokové schéma komunikace mezi komponentami a směrovači	21
5	Náhled vrstev aplikace	22
6	Sekvenční diagram - proces přihlášení	23
7	Pohled na třídu prostřednictvím které se webová služba přihlašuje ke směrovačům	23
8	Zachycení celé komunikace od webového rozhraní až po směrovač	27
9	Třídní diagram webové služby	28
10	Dodržování pravidel provozu	29
11	Aplikace tvarování datového toku	30
12	Ukázková topologie	31
13	Sekvenční diagram vložení pravidla fronty	33
14	Sekvenční diagram načtení pravidel fronty	34
15	Sekvenční diagram smazání pravidla fronty	34
16	Výsledný graf vytížení linky	38
17	Diagram aktivit popisujících chování PHP skriptu	39
18	Prezentace systémových zdrojů	43
19	Diagram případu užití webového rozhraní	45

Seznam výpisů zdrojového kódu

1	Spuštění naslouchání komunikace přes Mikrotik API	18
2	Porovnání zápisu Mikrotik API a SSH	25
3	Odpověď získaná pomocí API	25
4	Výsledek pomocí SSH příkazu z výpisu 2	25
5	Zpracování odpovědi z výpisu 3	26
6	Ukázka přidělení šířky pásma	32
7	Vytvoření RRD databáze	36
8	Občerstvení RRD databáze	36
9	Vytvoření grafu	37
10	Konfigurace periodického spouštění skriptu	40

1 Úvod

Při budování počítačové sítě je důležité mít neustálý přehled o síťových prvcích, které jsou nasazeny v počítačové síti. Předně je důležité, aby správce, který je zodpovědný za funkčnost počítačové sítě, měl neustálý přehled o dění v počítačové síti. Pro základní přehled může stačit informace jestli prvek funguje, či nikoliv. Taková informace je sice hodně důležitá, ale v momentě kdy síť přestane fungovat je velice často pozdě. Náprava vzniklého problému v situaci s nedostatkem informací je většinou drahá, provizorní a představuje další nepřímé finanční ztráty pro poskytovatele. Provizorní řešení je potřeba později vyřešit řešením stálým, třeba že to znamená další výpadek, či omezení služeb [1].

Firemní počítačová síť již dnes neposkytuje pouze přístup do internetu. Firemní síť se dnes velice často využívá k telekomunikačním službám [11] a komunikaci [1] v rámci firmy či jejich poboček. Jedná-li se o počítačovou síť střední nebo velké firmy s řádově stovkami účastníků u nichž je předpoklad, že při své práci využívají firemní počítačovou síť, mohou být tyto ztráty značně vysoké.

Samostatnou kapitolou jsou firmy zabývající se poskytováním pronájmu datových kapacit svých počítačových sítí [4]. Kapacity pronajímají k hlasovým a datovým službám, zprostředkovávají připojení firem nebo koncových účastníků k síti internet. Výpadek nebo omezení služby může mít mimo aktuálních ztrát i negativní ohlas v podobě budoucího neúspěšného výběrového řízení či odlivu zákazníků ke konkurenci.

Tyto firmy se při provozování své počítačové sítě nespokojí se základními informacemi při monitorování své infrastruktury a požadují více informací. Příkladem detailních informací je informace o propustnosti spojení mezi konkrétními body počítačové sítě, zpoždění při komunikaci mezi sousedními body nebo vytížení konkrétních spojení mezi body v počítačové síti. Firmám, využívajícím ve své počítačové síti mikrovlnné spoje ve standardu WiFi [25], mohou být přínosem i informace o stavu rádiových rozhraní. Tyto firmy provozují své služby často v českých městech a obcích, kde poskytují přístup do sítě internet pro koncové uživatele prostřednictvím bezdrátové technologie WiFi [25]. Na základě získání a zpracování takovýchto dat lze vhodně načasovat odstávku části sítě a provést její údržbu. Vhodně načasovanou modernizací a vylepšením síťové infrastruktury, lze lépe optimalizovat výdaje na modernizaci síťových komponent a správu celé počítačové sítě.

Získaná data ze síťových prvků, je potřeba zpracovat a předat správci zodpovědnému za chod počítačové sítě. K monitorování a správě síťových prvků slouží uživatelské rozhraní, prostřednictvím kterého lze síťové zařízení ovládat a nastavit. Pokud máme v síti více zařízení, je monitorování a sledování změn v síti velice obtížné. Zařízení používaná v počítačové síti mohou mít různá uživatelská rozhraní. Je proto velice žádoucí přenést informace ze síťových prvků na jedno místo, odkud se mohou prezentovat správci sítě pomocí jednoho uživatelského rozhraní. Tento postup šetří čas a zpřehledňuje dění na počítačové síti. Na základě kvalitních a rychlých informací může správce operativně řešit případné problémy.

Téma diplomové práce, zabývající se vzdálenou správou síťových prvků, se zrodilo při konzultaci se zástupci firmy, která poskytuje přístup do sítě internet ve slezských

městech a obcích. Firma by ráda vylepšila svůj vlastní informační systém o možnost sledování dění na počítačové síti a to prostřednictvím totožného webového rozhraní, které obsluhu informačního systému zná.

1.1 Struktura diplomové práce

Diplomová práce obsahuje teoretickou i praktickou část.

Teoretická část diplomové práce se v kapitole 2 zabývá analýzou stávajících řešení a možnostmi správy síťové infrastruktury. Návrhem, na vylepšení stávajícího řešení v oblasti správy síťové infrastruktury, se zabývá kapitola 3. Popis procesu komunikace se směrovači je popsán v kapitole 4, kde jsou uvedeny i praktické ukázky komunikace se směrovačem. Rozhraní, které spravuje kapacity síťové infrastruktury, je popsáno v kapitole 5. V kapitole je také nastíněn teoretický úvod do problematiky řízení provozu. Kapitola 6 poskytuje náhled na problematiku tvorby grafu a sledování proměnných v čase. Popisem sledování síťových komponent se zabývá kapitola 7. Návrh webového rozhraní, které demonstrovuje funkce webové služby, je popsán v kapitole 8. Kapitola 9 se popisuje nástroje, který zobrazují topologii sítě a její evidenci.

Praktická část diplomové práce se zabývá implementací řešení správy síťové infrastruktury. Výsledná podoba implementace je uložena na přiloženém CD.

2 Přehled aplikací monitorujících síťové prvky

Přehled charakterizuje aplikace určené k monitorování síťových prvků. Aplikace jsou používány firmami poskytujícími internet na Opavsku. Jedná se jednak o aplikace volně dostupné na internetu, ale i o komerční programy. Mezi aplikacemi sloužícími k monitorování síťových prvků jsou aplikace, které slouží ke sběru informací na počítačové síti a také ty, které se zaměřují na grafickou vizualizaci počítačové sítě. Sbírané informace jsou následně, prostřednictvím aplikace, využity k prezentaci pomocí uživatelského rozhraní.

2.1 Nagios

Prvním zástupcem monitorovacích aplikací je Nagios [16]. Jde o aplikaci, jež se do roku 2002 jmenovala Netsaint. Používá se jako automatizované sledování stavu počítačové sítě a služeb, běžících na počítačové síti. Aplikace je primárně vyvíjená pro operační systém Linux, ale lze jej provozovat i na jiných operačních systémech Unixového typu. Aplikace je vydávána pod svobodnou licencí GPL [2]. Lze ji tedy použít i pro komerční účely a není třeba si zakupovat licenci. Monitorovat lze služby jako SMTP, POP3, HTTP, NNTP, ICMP, SNMP. V případě výskytu problému umožňuje Nagios zaslat oznámení správci prostřednictvím e-mailu, sms nebo pomocí služeb rychlého doručování zpráv jako je ICQ.

2.1.1 Výhody

Nespornou výhodou jsou příznivé licenční a finanční podmínky, za kterých je možno Nagios používat. Pořizovací náklady jsou nulové a operační systém Linux, který se nejčastěji používá k běhu aplikace, lze pořídit také s nulovými náklady. Nagios umí monitorovat různé operační systémy napříč hlavními platformami Windows, Unix a Linux. Na těchto systémech umí monitorovat dostupnost síťových služeb. V případě nutnosti lze Nagios rozšířit o vlastní moduly naprogramované v jazycích Bash, C++, Perl, Python a další [3].

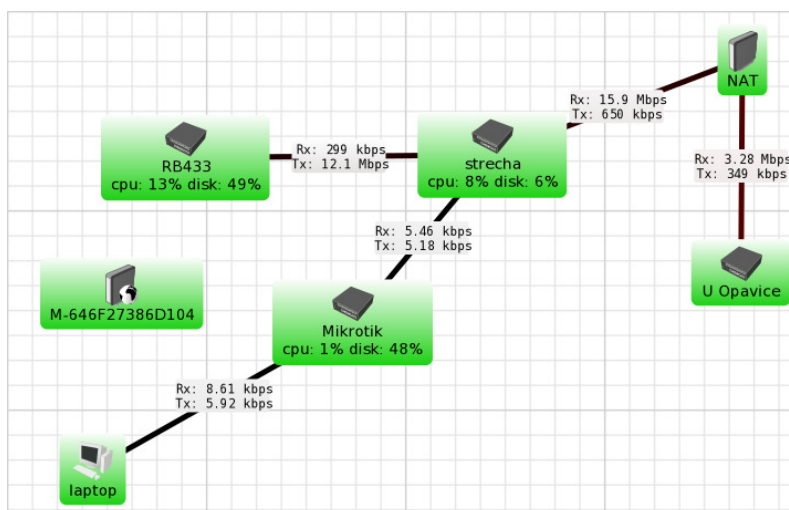
2.1.2 Nedostatky

Mezi nedostatky lze považovat velké nároky na zdroje, jimiž musí disponovat server na kterém Nagios běží. Jedná se především o množství operační paměti. Konfigurace aplikace Nagios se provádí ručně, což může být pro někoho výhodou pro jiného nevýhodou. Samotná konfigurace je časově velmi náročná.

2.2 The Dude

The Dude [23] je aplikace dostupná pro platformu Windows. Aplikaci The Dude lze volně využívat. Nejedná se však přímo o monitorovací aplikaci, která by shromažďovala data ze síťových prvků a ty následně prezentovala, ale o aplikaci umožňující vytvořit schéma reálné počítačové sítě. Za pomoci síťových protokolů umožňuje tato aplikace načíst nebo nastavit vlastnosti síťových prvků. Takto vytvořená počítačová síť je vlastně graf, kde vrcholy grafu jsou směrovače (síťové prvky) a hrany grafu linky propojující směrovače.

Výrobcem této aplikace je firma Mikrotik, která se zabývá vývojem směrovacího operačního systému RouterOS [22], který vychází z operačního systému Linux. Z uvedeného je zřejmé, že se aplikace primárně zaměřuje na podporu směrovacích zařízení, na kterých běží tento operační systém. Se síťovými prvky Mikrotik umí Dude komunikovat pomocí protokolu SNMP nebo protokolu Mikrotik. U prvků jiných výrobců je komunikace omezena pouze na protokol SNMP.



Obrázek 1: Ukázka aplikace The Dude

The Dude umožňuje zobrazit aktuální zatížení síťových rozhraní mezi směrovači a na základě tohoto zatížení vykreslit graf. Na základě realizované komunikace mezi směrovači lze vybrat typ spojení, podle toho o jaký druh spojení se jedná (FastEthernet [5], WiFi 802.11a/g/b [25]). Spojení mezi směrovači zaznamenané v aplikaci Dude se dle aktuálního zatížení konkrétního spojení zabarvuje. Má-li spojení dostatek volné kapacity je spojení zbarveno černě. Pokud se zatížení spojení blíží maximálně dostupné kapacitě, změní se barva spojení na červenou.

Pokud potřebuje správce počítačové sítě upravit vlastnost konkrétního směrovače¹, lze využít přímo rozhraní aplikace The Dude. V případě většího zásahu na směrovači lze

¹Směrovač musí běžet na platformě Mikrotik

v aplikaci The Dude spustit konfigurační aplikaci Winbox². Za pomoci protokolu SSH nebo Telnet lze využít další možnost vzdálené správy směrovače.

Aplikace The Dude může pracovat ve dvou základních režimech a to v režimu server a v režimu klient. V režimu server se vytvoří (nakreslí) celá topologie počítačové sítě. Předpokládá se, že serverová část bude spuštěna neustále a tím bude občerstvovat data ze síťových prvků sledované počítačové sítě. V režimu klient se The Dude připojí na server, z kterého si načte topologii sledované sítě a aktuální data.

Pokud má k topologii přístup více správců, lze omezit přístup k aplikaci pomocí uživatelského jména a hesla. Každý uživatel může mít různá práva k přístupu a nastavení.

2.2.1 Výhody a nedostatky

Správce má vizuální přehled o počítačové síti, včetně informací jak jsou jednotlivá spojení zatížená. Správce může pohotově řešit aktuální problémy. Pokud se o síť stará více správců, lze jim přidělit různá práva a omezení přístupu.

Často se stává, že aplikace přestane reagovat a dojde ke ztrátě dat. Aplikace je dostupná pouze pro platformu Windows.

2.3 ISPadmin

Jde o komerční aplikace umožňující monitorovat a nastavovat vlastnosti u síťových prvků. Aplikace podporuje správu síťových prvků běžících na operačním systému Linux a Mikrotik Router OS [22]. Ovládací rozhraní je řešeno pomocí internetového prohlížeče. Vývoj, technickou podporu a distribuci, zajišťuje firma Net service solution s.r.o. sídlící v Šumperku. ISP admin [10] je zaměřen na firmy poskytující datové, hlasové a multimediální služby koncovým zákazníkům z řad domácností, malých a středních firem. Z tohoto zaměření pramení i další služby, které aplikace nabízí [10]. Klíčové služby které aplikace poskytuje:

2.3.1 Evidence zákazníků připojených do sítě

Jedna se základních funkcí je evidence a správa klientů/zákazníků. K zákazníkovi se vážou služby jež má nastavené a k nim zvoleny tarify. Dle zvoleného tarifu se generují faktury a smlouvy. U evidovaného zákazníka lze zobrazit podrobnější informace o počtu přenesených dat z/do internetu. Dále u zákazníka evidujeme technické informace jako je šířka pásma vedoucí od zákazníka do internetu a opačně [10].

2.3.2 Fakturační systém

Fakturační systém generuje faktury pro zákazníky automaticky na základě poskytovaných služeb a vybraných tarifů. Zákazníkům je umožněno stažení faktury za určité kalendářní období. Ověření příchodu platby za poskytnutou službu na účet poskytovatele.

²Aplikace, která slouží k nastavování a ovládání směrovačů na platformě Mikrotik RouterOS pomocí přehledného GUI

Na základě stavu faktury (uhrazena - neuhrzena) může systém vygenerovat pravidla pro síťové prvky:

- Omezení služby v podobě snížení šířky pásma.
- Omezení vybraných internetových služeb.
- Odepření využití objednaných služeb.

2.3.3 Evidence směrovačů a modemů

ISPadmin umožňuje evidovat směrovače nebo modemy v počítačové síti poskytovatele. Evidují se také informace důležité k zajištění monitorování a správě směrovačů. ISP admin podporuje tyto typy [10] směrovačů:

- Mikrotik
- Flash Linux router
- Ubiquiti
- Canopy

2.3.4 Další funkce aplikace ISPadmin

Velkým pomocníkem při odhalování problémů je možnost tvorby statistik síťových veličin. Statistiky se tvoří pomocí grafů, kde jsou síťové veličiny zaznamenávány v čase.

K přehledu o výlukách, o práci techniků a o řešení problémů slouží modul plánování. Tento modul aplikace slouží jako jednotné místo kde se shromažďují poruchy, požadavky od zákazníků nebo plánování nového rozšiřování počítačové sítě. K těmto jednotlivým záznamům se přihlašují technici, kteří vzniklé problémy řeší [10].

Pro potřeby zaznamenávání provozu sítě slouží modul s názvem *netflow* [17]. Modul sbírá a zobrazuje informace o probíhající komunikaci zákazníků se servery v internetu [10]. Informace se získávají ze záhlaví protokolu IP. Informace o provozu se posílají z hraničního směrovače, který je před směrovačem s funkcí NAT[17]. Zaznamenávání provozu bylo se zákona povinné pro poskytovatele veřejně dostupných služeb. Poskytovatelé veřejně dostupné služby museli tyto záznamy uchovávat po dobu šesti měsíců. Informace měly sloužit pro potřeby orgánům činným v trestním řízení při vyšetřování.

Poznámka 2.1 Ústavní soud zrušil povinnost uchovávat data o provozu ve veřejně dostupných službách dne 22. března 2011 [24]. Tato povinnost se netýkala jen poskytovatelů internetu, ale například i mobilních operátorů.

V aplikaci je umožněno, dle zvolené sekce, filtrování zákazníků podle poskytované služby, poskytovaného tarifu, fakturačního období, případně fulltextového vyhledávání.

2.3.5 Výhody

ISP admin řeší většinu agend spojených s poskytováním internetu koncovým zákazníkům. Poskytovatel dostane aplikaci, která za něj řeší většinu problémů, které jsou spojeny s poskytováním služeb. Poskytovatel se tak může věnovat jiným věcem. Aplikaci je možno vyzkoušet zdarma na webových stránkách výrobce [10] a je možno její propojení s účetními systémy.

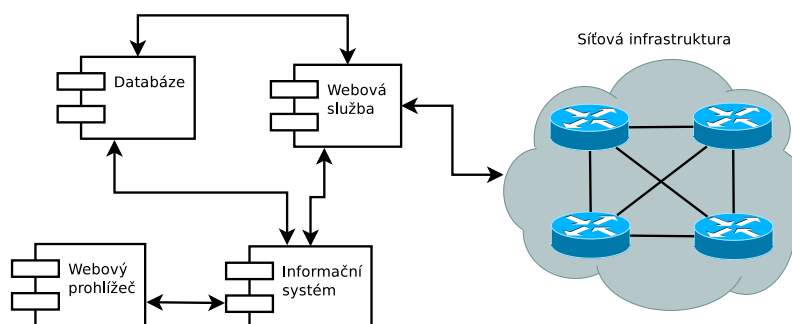
2.3.6 Nedostatky

Nedostatky jsou spíše subjektivní záležitostí. Hlavním problémem je především počáteční přechod z již fungujícího informačního systému na tento komerční celek.

- Uzavřenost aplikace.
- Aplikace jako celek, nelze oddělit monitoring a sběr dat od jejich prezentace.

3 Analýza a popis vlastního řešení správy síťové infrastruktury

Cílem diplomové práce bylo navrhnout vlastní řešení správy síťové infrastruktury. Nově navržené řešení správy síťové infrastruktury umožní jinou formu komunikace se síťovou infrastrukturou. Klíčovým prvkem navrženého řešení bude webová služba, která nabídne, za pomoci vzdáleného volání procedur, ovládání směrovače. Možnost vzdáleného volání procedur umožní přístup do síťové infrastruktury a tím zprostředkuje správu směrovače přes různá uživatelská rozhraní. Cílem této diplomové práce tedy nebylo budovat nový informační systém, ale nabídnout komunikační rozhraní se směrovači.



Obrázek 2: Náhled na návrh aplikace

3.1 Nové funkce stávajícího informačního systému

Vychází se s předpokladu, že konkrétní poskytovatel již má informační systém ve kterém eviduje zákazníky a tento systém by mohl rozšířit o nové vlastnosti bez zásahu do fungující části. Může se jednat o vlastní informační systém pro evidenci zákazníků, generování faktur za určité období apod. Tento stávající informační systém by firma ráda rozšířila například o statistiky stažených dat za dané období, o vytížení komunikačních linek mezi směrovači nebo o možnost správy směrovače.

3.2 Charakteristika navrhovaného řešení

Prvním úkolem je shromáždit data a nabídnout je ve srozumitelné formě k dalšímu zpracování. Druhým úkolem je vytvořit takovou možnost, aby hodnoty ovlivňující chování směrovačů bylo možné zadávat pomocí stávajícího rozhraní informačního systému.

3.2.1 Komunikace mezi webovou aplikací a směrovači

Směrovače, na kterých se prováděly testy, jsou uvedeny v tabulce 1. Komunikace, probíhající mezi směrovači a webovou službou, probíhá pomocí Mikrotik API [13]. Toto API umožňuje jak přijímat data ze směrovače tak do něj data (konfigurační respektive diagnostické příkazy) posílat. Pro přístup stačí na směrovači povolit tuto možnost komunikace.

Výpis 1 obsahuje příkaz, kterým lze povolit naslouchání komunikace prostřednictvím Mikrotik API. Obrázek 2 ilustruje schéma aplikace a cestu komunikace.

Poznámka 3.1 Mikrotik API je součástí operačního systému Router OS až od verze 3.0 a výše

```
ip service enable api
```

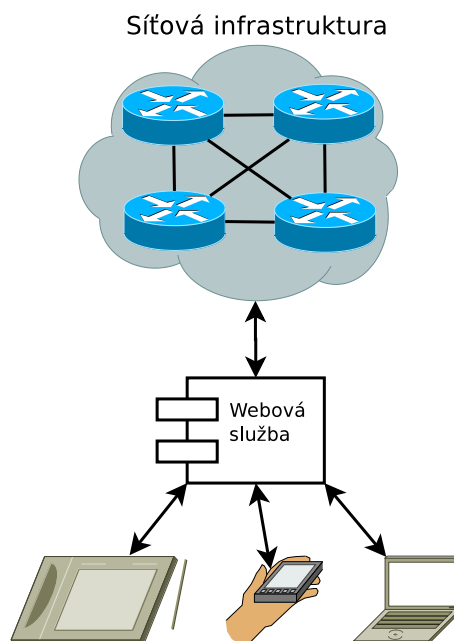
Výpis 1: Spuštění naslouchání komunikace přes Mikrotik API

Operační systém	verze	Platforma
Mikrotik RouterOS	3.22	x86 - VirtualBox
Mikrotik RouterOS	5.4	MIPS - RB750

Tabulka 1: Tabulka operačních systémů běžících na použitých směrovačích

3.2.2 Získaná data

Data získaná ze směrovačů prostřednictvím Mikrotik API jsou ve tvaru prostého textu ekvivalentní k datovému typu `string`. Získaná data jsou ve tvaru klíč hodnota a lze je zpracovat datovou strukturou `slovník`. Problematicke se podrobně věnuje následující kapitola 4.



Obrázek 3: Přístup z různých druhů zařízení

3.3 Oddělení sběru dat od jejich prezentace

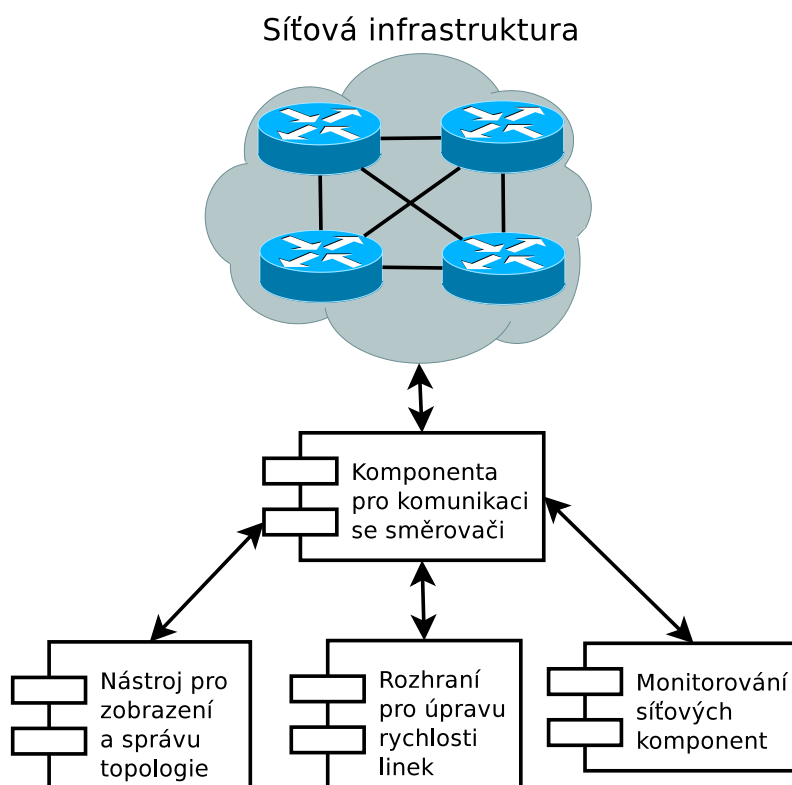
Vhodným řešením, které splňuje požadavky na oddělení sběru dat z počítačové sítě od jejich prezentace, je vytvoření webové služby. Webová služba bude mít za úkol komunikovat se směrovači v počítačové síti a bude fungovat jistým způsobem jako překladová tabulka mezi funkcemi vzdáleně volanými z programu a příkazy zasílanými na směrovač. Blokové schéma popisující tento proces nabízí obrázek 2. Řešení navržené pomocí webové služby umožní komunikovat se směrovači v počítačové síti s informačními systémy napříč spektrem programovacích jazyků.

3.4 Výhody navrhovaného řešení

Vedlejším efektem při použití webové služby (i když v tomto případě příznivý) bude aplikace, jejíž prezentační část nebude závislá na platformě. Do budoucna se nemusí vývojáři zaměřovat pouze na získávání dat prostřednictvím webového rozhraní. Přístup k webové službě se může rozšířit například o přístup prostřednictvím mobilního telefonu, tabletu apod. Viz obrázek 3.

4 Komponenta pro komunikaci se směrovači

Komponenta, která zajišťuje komunikaci se směrovači, je na celém projektu nejdůležitější. Prostřednictvím této komponenty se bude provádět veškerá komunikace mezi aplikací a směrovači. Na obrázku 4 je znázorněno blokové schéma probíhající komunikace. Komponenta pracuje na nejnižší vrstvě aplikace. Obrázek 5 znázorňuje rozvrstvení aplikace.

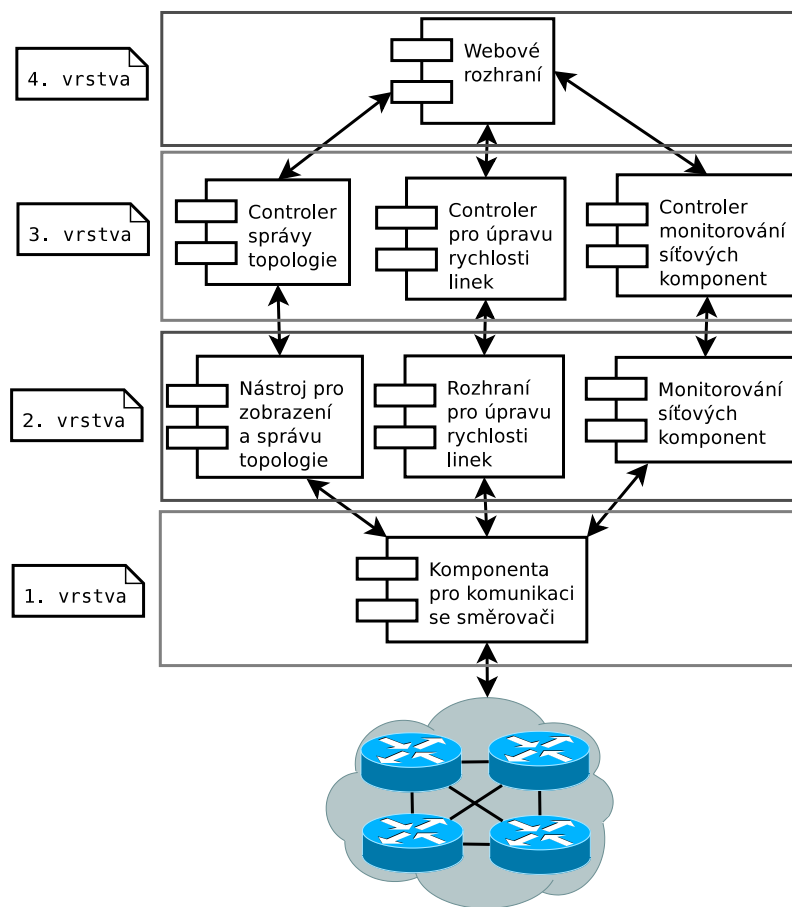


Obrázek 4: Blokové schéma komunikace mezi komponentami a směrovači

4.1 Popis komponenty

Komponenta představuje překladovou tabulku mezi uživatelským rozhraním a směrovači. Úkoly této komponenty jsou sepsány v následujících bodech.

1. Navázat spojení s konkrétním směrovačem.
2. Poslat příkaz směrovači obdrženy z vyšší vrstvy aplikace.
3. Načíst případnou odpověď od směrovače.
4. Odpověď převést z textové podoby do objektové.
5. Předat objekt, respektive kolekci objektů, vyšší úrovni aplikace.



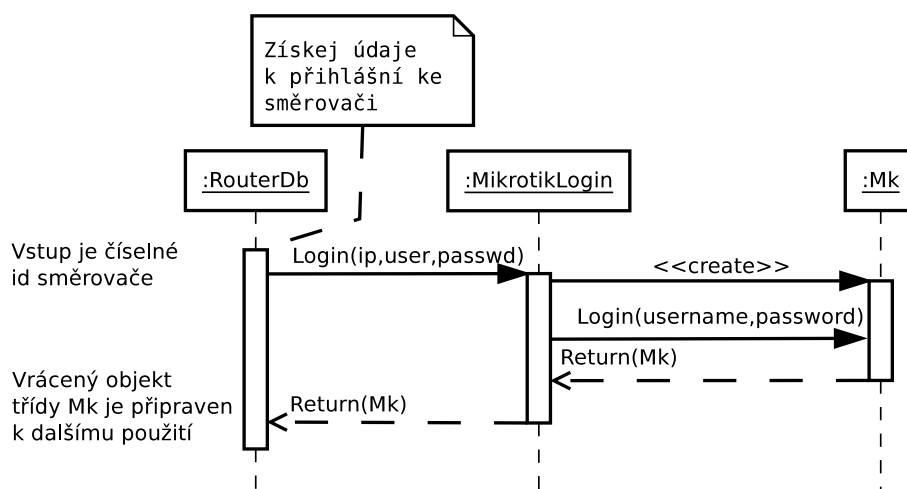
Obrázek 5: Náhled vrstev aplikace

4.2 Přihlášení ke směrovači prostřednictvím Mikrotik API

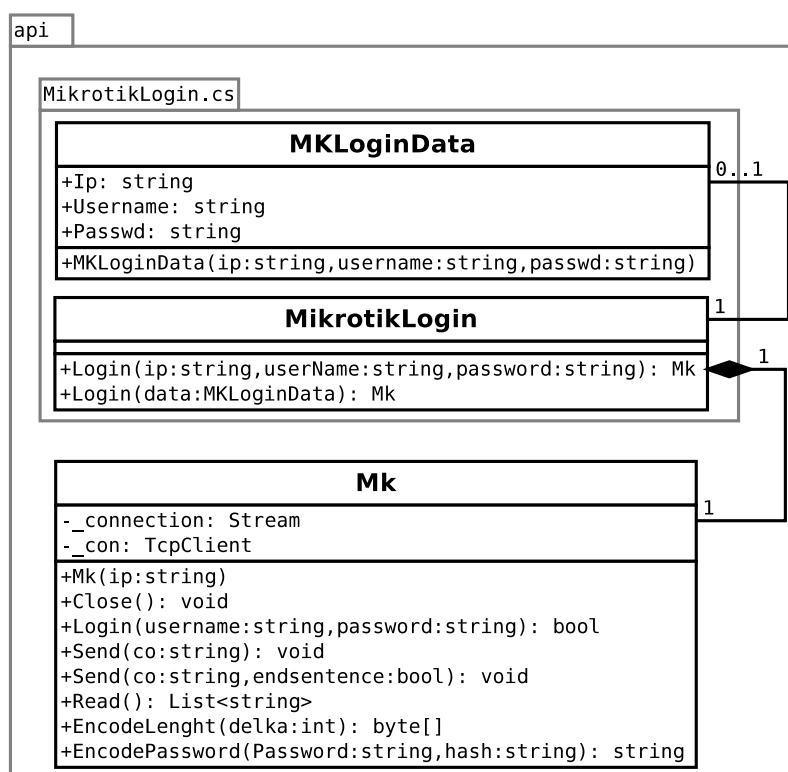
Před zahájením komunikace se směrovači, je potřeba se k jednomu konkrétnímu přihlásit. Přihlášení proběhne pomocí funkce *Login()*, která se nachází ve třídě *Mk*. Pro potřebu ošetření vstupu a řízeného přihlášení byla vytvořena třída *MikrotikLogin*. Propojení tříd je znázorněno na obrázku 7 v podobě třídního diagramu. Samotný proces přihlášení je zachycen sekvenčním diagramem na obrázku 6. Třída *MikrotikLogin* má za úkol:

1. Přihlásit se ke směrovači pomocí funkce *Login()* ve třídě *Mk*.
2. Po úspěšném přihlášení vrátí objekt třídy *Mk*.

Získaný objekt třídy *Mk* je přihlášen ke konkrétnímu směrovači a je připraven k další práci se směrovačem. Přihlášení ke směrovači je následně využito při posílání a načítání případné odpovědi od směrovače.



Obrázek 6: Sekvenční diagram - proces přihlášení



Obrázek 7: Pohled na třídu prostřednictvím které se webová služba přihlašuje ke směrovačům

4.3 Popis třídy Mk

Třída `Mk` je převzatá z wiki stránek firmy Mikrotik sloužící k podpoře směrovačů a operačního systému. Odkaz na API je uveden v referencích [12]. Grafická interpretace třídy `Mk`, formou třídního diagramu, je zachycena na obrázku 7. Třída je připravena zahájit komunikace se směrovačem, zasílat či přijímat odpovědi zaslané směrovačem. Po dokončení komunikace spojení zanikne.

Jedná se o dynamickou třídu a je potřeba vytvořit objekt této třídy. Konstruktor akceptuje vstupní parametr typu `string`, kterým se definuje IP adresa cílového směrovače. K přihlášení na směrovač je potřeba zavolat funkci `Login`. Funkce `Login` požaduje dva vstupní parametry datového typu `string`. Vstupními parametry jsou uživatelské jméno a heslo. Heslo je zadáváno v nezabezpečené podobě. Hash otisk udělá sama funkce `Login` při svém vykonávání.

Poznámka 4.1 Z důvodů evidence většího množství směrovačů ve sledované počítačové síti je důležité vhodně nastavit bezpečnost na směrovačích. Z důvodu nutnosti evidovat hesla v nezabezpečené podobě je vhodné vytvořit dalšího uživatele na směrovači, který bude mít povolen přístup pouze přes API rozhraní a to jen z konkrétní IP adresy (případně rozsahu). Také není vhodné, aby tento uživatel měl práva „full control“.

4.4 Analýza komponenty

Komponenta bude kompletně postavena na komunikaci prostřednictvím protokolu Mikrotik API. Hlavní funkce komponenty:

1. Přijímat příkazy od jiných komponent.
2. Pomocí vestavěných funkcí zajistit čtení dat ze směrovače.
3. Načtená data zpracovat a poslat komponentě, která si o data řekla.
4. Pomocí vestavěných funkcí poslat příkaz konkrétnímu směrovači.

Logický náhled popisuje funkcionalitu systému, kterou poskytuje webová služba. Funkce, které webová služba nabízí, ilustrují třídní diagram. Třídní diagram je zachycen obrázkem 9 na stránce 28.

Komunikaci mezi objekty popisuje sekvenční diagram. Sekvenční diagram zachycuje vytvoření požadavku prostřednictvím webové aplikace a jeho následné předání webové službě. Sekvenční diagram je zobrazen na obrázku 8 na stránce 9.

4.5 Posílání příkazů směrovači

Před samotným stažením libovolných informací je potřeba směrovači zadat příkaz o požadovaná data. Příkazy se zapisují ve tvaru vhodném pro Mikrotik API, tvar je podobný příkazům, které se posílají směrovači například pomocí SSH, či telnetu. Níže jsou uvedeny obě formy zápisu.

Poznámka 4.2 Na prvním řádku je zápis Mikrotik API, na třetím pak obdobný zápis pomocí SSH

-
1. /system/resource/print
 - 2.
 3. system resource print
-

Výpis 2: Porovnání zápisu Mikrotik API a SSH

Příkazem zmíněným ve výpisu 2 nám směrovač pošle informace o sobě. Jsou to informace jako vytížení procesorů, počet procesorů, typ procesoru, využití paměti apod. Ve výpisech 3 a 4 jsou uvedeny odpovědi, které směrovač odpoví.

```
!re=uptime=04:04:27=version=3.22=free-memory=48236=total-memory=57612=cpu=Intel(R)=cpu
-count=1=cpu-frequency=1274=cpu-load=1=free-hdd-space=462831=total-hdd-space
=497357=write-sect-since-reboot=351=write-sect-total=351=architecture-name=x86=
board-name=x86
!done
```

Výpis 3: Odpověď získaná pomocí API

4.5.1 Získaná data prostřednictvím Mikrotik API

Z výpisu 3 je vidět, že s pomocí Mikrotik API se odpověď získává v podobě řetězce znaků. Při práci s tímto řetězcem je potřeba oddělit užitečné informace od neužitečných.

```
uptime: 4h7m41s
version: "3.22"
free-memory: 48004kB
total-memory: 57612kB
cpu: "Intel (R)"
cpu-count: 1
cpu-frequency: 1274MHz
cpu-load: 2
free-hdd-space: 462831kB
total-hdd-space: 497357kB
write-sect-since-reboot: 351
write-sect-total: 351
architecture-name: "x86"
board-name: "x86"
```

Výpis 4: Výsledek pomocí SSH příkazu z výpisu 2

Poznámka 4.3 Problémem je, že různé verze operačního systému Router OS vrací poněkud odlišnou odpověď. Je potřeba zvolit vhodnou datovou strukturu k zachycení odpovědi např. slovník.

4.6 Obecná implementace

Získaná data jsou ve formě řetězce znaků, jak je popsáno ve výpisu 3. Dále je tento řetězec oddělen od nepotřebných znaků a uložen do datové struktury slovník. Datová struktura je definována tak, aby klíč i hodnota byly datového typu `string`. Slovník je vhodným řešením pro předávání výsledků napříč programovacími jazyky. Jedná se o jednu ze základních datových struktur.

Poznámka 4.4 Jednou z možností, jak do vestavěných komponent ASP.NET vložit a zobrazit data, je vkládat data v objektovém tvaru [15]. Slovník je proto vhodné předefinovat na objektovou datovou strukturu.

4.6.1 Náhled implementace

Výpis 5 nabízí pohled na zpracování odpovědi, kterou zaslal směrovač pomocí Mikrotoik API.

```
public static SystemResource GetResource(Mikrotik mikrotik)
{
    var res = new Dictionary<string, string>();

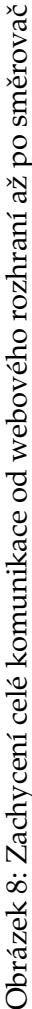
    mikrotik.Send("/system/resource/print", true);
    string[] tempVar = mikrotik.Read()[0].Split(' ');

    for (int i = 1; i < (tempVar.Length - 1); i += 2)
    {
        res.Add(tempVar[i], tempVar[i + 1]);
    }
    return RouterObjectMapping.GetResourceMap(res);
}
```

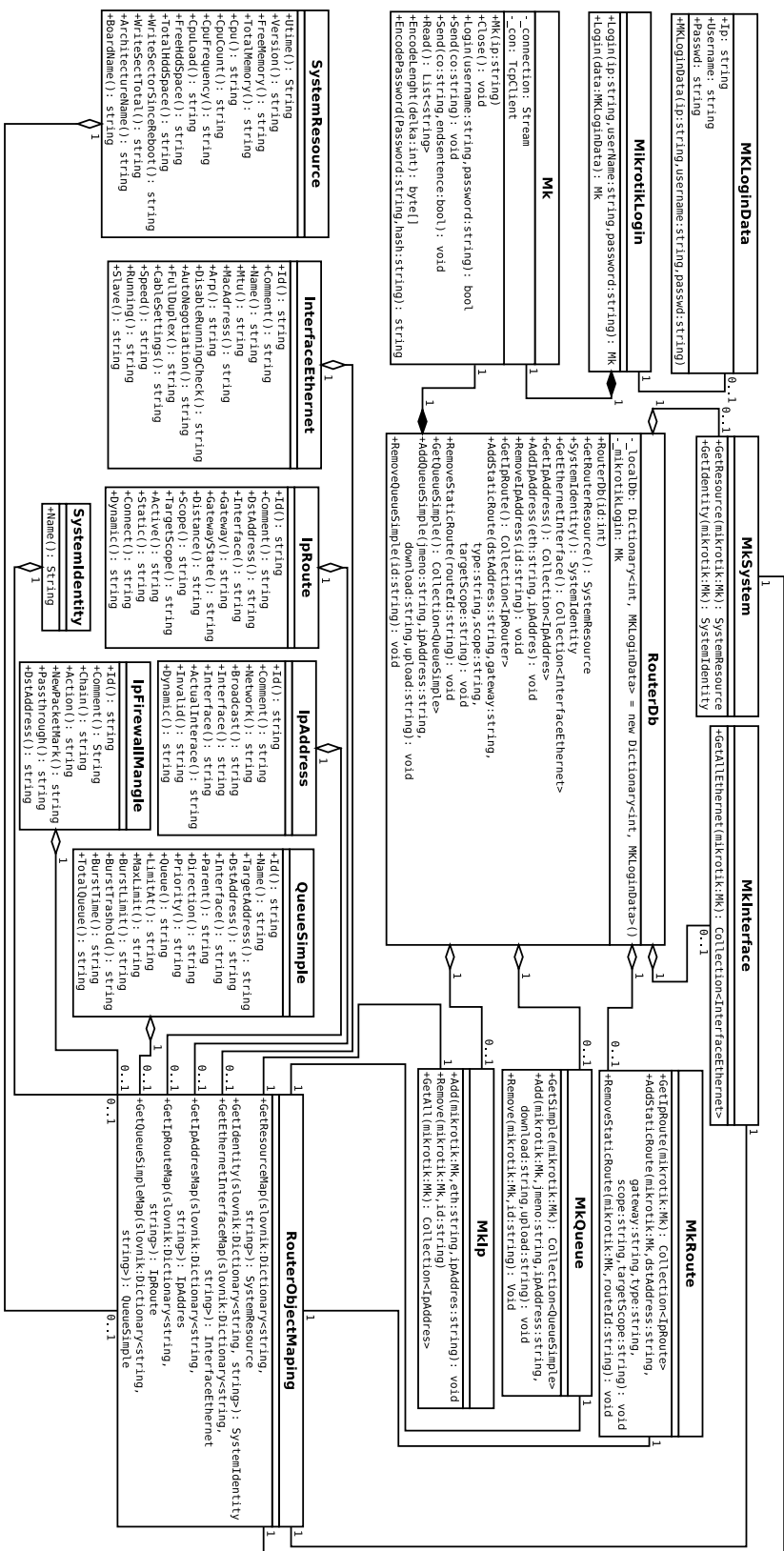
Výpis 5: Zpracování odpovědi z výpisu 3

4.6.2 Přetypování na objekty

K přetypování datové struktury slovník na objektový datový typ slouží třída *RouterObjectMapping*. Ve třídě jsou vnitřní funkce které akceptují na vstupu generickou datovou strukturu slovník. Slovník má klíč i hodnotu nastavenou na datový typ `string`. Výstupem z funkcí jsou objekty tříd popisující konkrétní návratovou hodnotu.



Obrázek 8: Zachycení celé komunikace od webového rozhraní až po směrovač



Obrázek 9: Třídní diagram webové služby

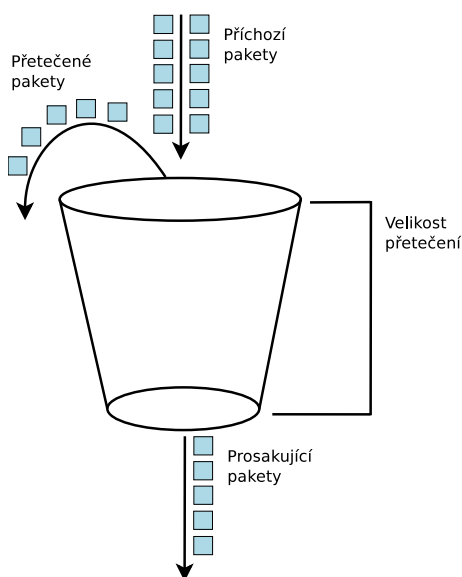
5 Rozhraní pro úpravu rychlosti linky

Kapacita linky k zákazníkovi, prostřednictvím počítačové sítě, má často větší kapacitu než si zákazník objedná/zaplatí. Důvodem jsou velké rozdíly mezi rychlostmi spojení odvozené od verze Ethernet [18]. Tabulka 2 popisuje verze Ethernet a rozdíly mezi rychlostmi.

Verze Ethernet	Rychlost spojení
Ethernet	10 Mbit/s
Fast Ethernet	100 Mbit/s
Gigabit Ethernet	1000 Mbit/s
Ten Gigabit Ethernet	10 Gbit/s

Tabulka 2: Verze Ethernetu a její rychlost

Nejběžnější způsob připojení do počítačové sítě je rychlost 100 Mbit/s na spojové vrstvě (*Fast Ethernet*). Zákazník si například objedná rychlost 20 Mbit/s. Je tedy potřeba zajistit jisté omezení šířky pásma na vyšší vrstvě ISO/OSI [8] modelu. O tento způsobu „omezení“ řízení provozu na konkrétní lince se stará *Traffic shaping* [20].



Obrázek 10: Dodržování pravidel provozu

5.1 Traffic shaping

Český název pro pojem *Traffic shaping* je řízení provozu někdy také označeno jako tvarování provozu. Jedná se v principu o zpožďování některých paketů ve frontě směrovače. Tyto pakety jsou často méně důležité. Jde o zabránění přetečení bufferu na

síťových rozhraních, kdy směrovač není schopen zpracovat tolik paketu, kolik se na něj posílá. Používá se také k dodržování pravidel, která jsou definována pomocí funkce *Traffic Policing*. V případě, že směrovač není schopen zpracovávat požadavky, odešle vysílací straně ICMP [17] zprávu. ICMP zpráva sdělí vysílací straně, aby omezila šířku vysílaných zpráv. Tento popsaný princip funguje pokud se jedná o spojově orientovanou komunikaci na transportní vrstvě - protokol TCP. Transportní protokol TCP vždy po odeslání paketu čeká na potvrzení jeho doručení. V případě, že odpověď nedostane vyšle paket znovu.

Tento princip nelze uplatnit pokud je na transportní vrstvě komunikováno prostřednictvím UDP protokolu. V případě, že se jedná o komunikaci na transportní vrstvě pomocí UDP a směrovač není schopen zpracovávat příchozí pakety, jednoduše je zahodí. [20]

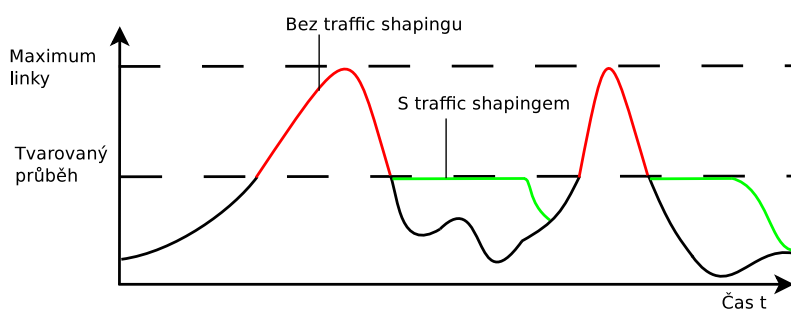
5.2 Traffic Policing

Jde o nastavení pravidel datového toku procházející směrovačem. Pravidlem je myšlena velikost datového toku probíhajícího přes směrovač. Obrázek 10 ilustruje dodržování pravidel datového toku. Pakety, které jsou mimo vyčleněnou šířku pásma, se zahazují.

5.3 Implementace tvarování provozu

Aby bylo zajištěno dodržování pravidel v datovém toku je zapotřebí upravit provoz tak, aby odpovídal pravidlům dané počítačové sítě. Proto se používá tvarování provozu. Na obrázku 11 je vidět jak se špičky provozu vytvarují. Červeně vyznačené špičky provozu jsou načteny do vyrovnávací paměti, kde jsou pozdrženy a postupně uvolňovány v čase. Proces tvarování provozu je znázorněn na obrázku zelenou barvou [5]. Tímto je zajištěno dodržování šířky pásma zákazníků na síti.

Při ukládání a čtení z vyrovnávací paměti dochází ke zpoždění kvůli čekání na vyrovnávací paměť. V případě že paměť dojde, nově příchozí data se zahodí. Koncovým stanicím nezbyvá než si o ztracená data znovu požádat³.

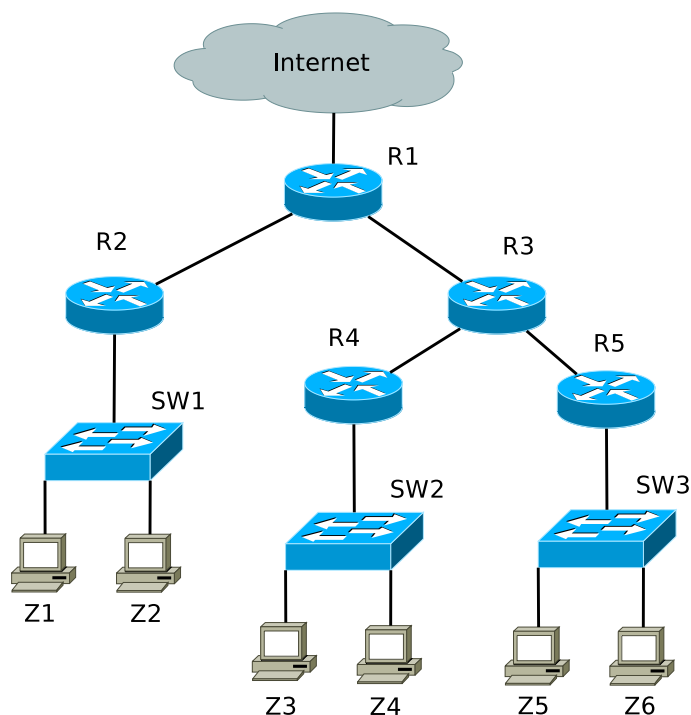


Obrázek 11: Aplikace tvarování datového toku

³Pokud se jedná o spojení pomocí protokolu TCP na transportní vrstvě

5.4 Řízení provozu na směrovači Mikrotik

V zařízeních Mikrotik slouží k jednoduché úpravě šířky pásma sekce *queue simple*. Úprava šířky pásma se provádí prostřednictvím směrovače nad oblastí, ve které chceme provést úpravu provozu.



Obrázek 12: Ukázková topologie

5.4.1 Umístění směrovače upravujícího rychlost pásma

Směrovač, který zajišťuje omezení šířky pásma provozu, jež vygeneruje zákazník, je vhodné umístit co nejbližší k zákazníkovi. Vhodnými kandidáty na úpravu šířky pásma od zákazníka jsou směrovače R2, R4, R5 zobrazené na obrázku 12. Tyto směrovače se nacházejí nejbližší koncovým stanicím.

Naopak provoz, který směřuje k zákazníkovi, je potřeba omezit co nejdále od zákazníka. Omezení provozu je žádoucí již při vstupu do počítačové sítě poskytovatele internetu. Nejvhodnějším kandidátem pro toto omezení provozu je směrovač R1 na obrázku 12.

5.4.2 Příklad omezení šířky pásma

Pokud je potřeba omezit provoz, respektive přidělit část pásma konkrétnímu zákazníkovi, je zapotřebí označit provoz procházející směrovačem. K identifikaci provozu stačí znát IP adresu koncové stanice nebo rozsah IP adres.

```
[admin@MikroTik] > queue simple add dst-address=10.0.0.4 max-limit=1M/1M
```

Výpis 6: Ukázka přidělení šířky pásma

Ve výpisu 6 je demonstrován příkaz k omezení šířky pásma pro zákazníka majícího IP adresu 10.0.0.4 na velikost 1 Mbit/s v obou směrech. Takový příkaz je možné směrovači zaslat prostřednictvím telnetu, či ssh konzole.

5.5 Implementace - webové rozhraní

Výše byly definovány požadavky, které jsou důležité k funkčnímu přidělení šířky pásma. K vytvoření nového pravidla na směrovači je zapotřebí vstup od správce systému. Důležité veličiny jsou IP adresa cílové stanice a velikosti šířky pásma od zákazníka směrem do internetu a šířka pásma pro opačnou cestu. Vstup se zadává prostřednictvím webového rozhraní. Popisem webového rozhraní se zabývá kapitola 8. Po obdržení vstupu se vytvoří příkaz, který se pošle pomocí komponenty, která má na starosti komunikaci se směrovači.

5.6 Návrh ovládaní rozhraní pro řízení rychlosti linky

Webové rozhraní umožňuje řízení rychlosti linky mezi zákazníkem a internetem. Webové rozhraní bude nabízet možnost prohlédnout stávající pravidla vytvořená na směrovači a nabídnout je správci v přehledné formě. Webová služba dále umožní vytvoření nového pravidla na směrovači a nepotřené pravidlo smazat.

5.6.1 Vkládání a odstranění pravidel

K vložení nového pravidla bude sloužit webový formulář. Webový formulář bude požadovat vložení údajů potřebných k vytvoření pravidla. Tyto údaje jsou:

- Textový popis pravidla potřebný k lepší identifikaci pravidla.
- Cílová IP adresa zákazníka.
- Velikost přidělené šířky pásma směrem k a od uživatele.

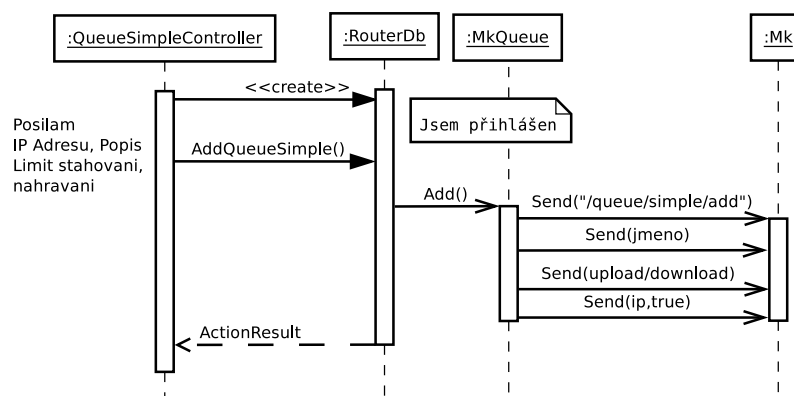
Odstranění pravidla bude umožněno, za pomoci příslušného tlačítka, přímo ze stránky přehledu pravidel na směrovači.

5.7 Implementace úpravy rychlosti linky - webová služba

Řízení pravidel fronty na směrovači má na starosti třída `MkQueue`. Třída má funkce obsluhující vkládání, mazání a načítání uložených pravidel.

5.7.1 Vložení nového pravidla

Vložení nového pravidla na směrovač ilustruje sekvenční diagram na obrázku 13. `QueueSimpleController` získá z webového rozhraní IP adresu, textový popis a limity pro nastavení šířky pásma z a do internetu. Tyto údaje pošle webové službě, prostřednictvím volání funkce `AddQueueSimple()` třídy `RouterDb`. Třída `RouterDb` drží ve vnitřní proměnné objekt s přihlášeným směrovačem. Tento objekt spolu s informacemi obdržnými od webového rozhraní vloží jako vstupní parametry funkci `Add()` ve třídě `MkQueue`. Funkce `Add()` odešle příkaz směrovači.



Obrázek 13: Sekvenční diagram vložení pravidla fronty

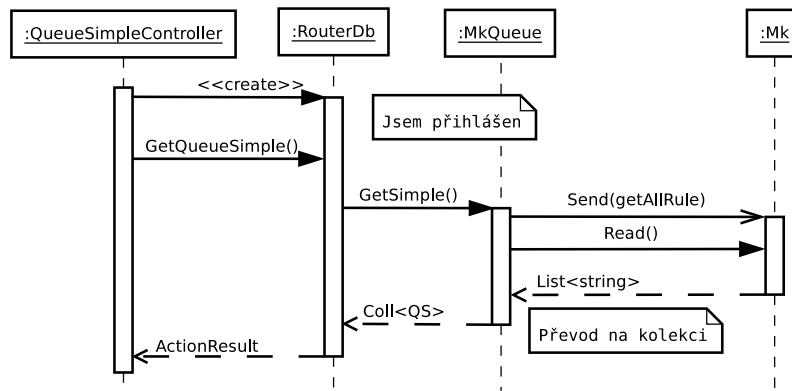
5.7.2 Načtení pravidel fronty

Přes webové rozhraní se pošle požadavek k načtení pravidel fronty. `QueueSimpleController` vytvoří instanci požadovaného směrovače a zavolá funkci `GetSimple()`. Funkce `GetSimple()` má vstupní parametr objekt, který je přihlášen ke směrovači. Tělo funkce `GetSimple()` provede odeslání příkazu na směrovač. Funkcí `Read()` se načtou pravidla fronty. Dále se ve funkci `GetSimple()` provede zpracování načtených pravidel do objektové podoby a vrátí se v kolekci objektu do `Controlleru`. `Controller` předá kolekci šabloně, která zajistí výpis pravidel na webové stránce webového rozhraní. Grafické znázornění je zachyceno sekvenčním diagramem na obrázku 14.

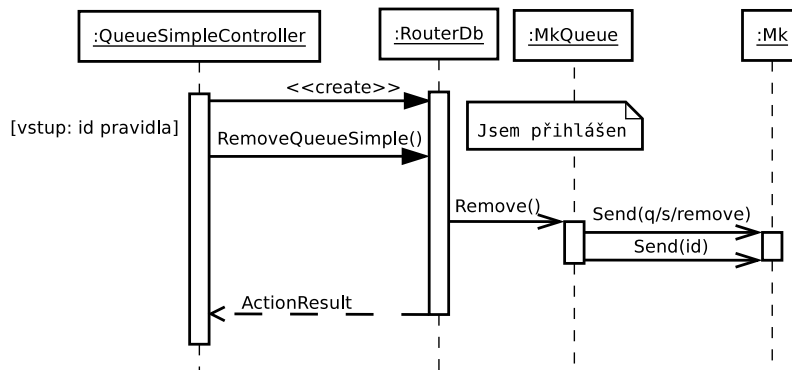
5.7.3 Odstranění pravidla fronty

Webová šablona předá `QueueSimpleControlleru` identifikátor pravidla, které je nadbytečné. `Controller` zavolá funkci `RemoveQueueSimple()` se vstupním parametrem identifikátoru pravidla určeného k odstranění. Funkce `RemoveQueueSimple()` zavolá funkci `Remove()`, kterou volá s parametry:

- Objekt s navázaným spojením na směrovač.
- Identifikátor pravidla určeného k odstranění.



Obrázek 14: Sekvenční diagram načtení pravidel fronty



Obrázek 15: Sekvenční diagram smazání pravidla fronty

Tělo funkce `Remove()` vyvolá dvakrát volání funkce `Send()`. V prvním volání pošle příkaz směrovači o tom, že je požadováno odstranění pravidla v sekci `queue simple`. Druhým voláním pošle funkce `Send()` identifikátor pravidla, které se má odstranit.

6 Tvorba grafů přenesených dat

Grafy jsou velice důležité při diagnostice problémů s počítačovou sítí. Řešiteli případného problému pomůže pohled na graf detekovat problém, který se na počítačové síti vyskytl nebo se periodicky vyskytuje. Sledované veličiny se zaznamenávají nejčastěji do 2D grafu s osami x a y , kde osa x znázorňuje časovou osu a osa y velikost sledované veličiny. Sledovat a zakreslit do grafu se dají různé síťové veličiny jako např:

- Propustnost přes síťové rozhraní.
- Odezva k cílové stanici/serveru/síťovému prvku.
- Chybovost na síťovém rozhraní.
- Aktivita zákazníků na počítačové síti.

Poznámka 6.1 Problémem vytvářením grafů z veličin vyskytujících se v počítačové síti se zabývá bakalářská práce v rámci odborné praxe [21]. Cílem této kapitoly je navázat na výsledky testování tvorby grafů, které jsou součástí bakalářské práce.

6.1 Tvorba grafů

Nástrojem pro tvorbu grafů byl zvolen program RRDtool. RRDtool je aplikace, která řeší pouze tvorbu grafů a k tomu nezbytné úkony. Za nezbytné úkony se dá považovat tvorba databáze a údržba aktuálních hodnot sledovaných veličin [19]. Program RRDtool má tři základní funkce:

1. Vytvoření databáze a definování veličin/ y .
2. Udržování databáze - obcerstvování hodnot sledovaných veličin.
3. Vykreslení veličin/ y do grafu.

Poznámka 6.2 Databáze je tvořena souborem uloženým na pevném disku. RRDtool si fyzickou organizaci řeší sám a proto není nutné vytvářet a udržovat klasický SŘBD [19].

Zaznamenat lze jakoukoli veličinu, která má velikost a lze ji vyjádřit číslem. Nemusí se jednat nutně o veličinu ze síťového prostředí. Je možné také zaznamenávat teplotu, využití operační paměti, využití procesorů a mnoho dalších.

6.2 Popis programu RRDtool

Program lze provozovat na platformách Windows, Linux a UNIX. Program nemá grafické rozhraní, pracuje se s ním pouze z příkazové řádky. Další možnost ovládní programu je využít knihoven k ovládní RRDtool v jazycích Perl, Python, Ruby a další.[19]

Typ sledované proměnné	Popis
Gauge	Používá se pro zápis aktuálních hodnot.
Counter	Používá se pro součtové hodnoty v čase.
Derive	Zaznamenává změnu mezi posledními dvěma záznamy.
Absolute	Použití najde v případě součtu hodnot v čase kdy načtená hodnota se nuluje.

Tabulka 3: Typ hodnot proměnných v databázi RRDtool

6.3 Použití programu rrdtool

Jak již bylo výše uvedeno, program se ovládá pomocí příkazové řádky příkazem `rrdtool`. Chování programu lze ovlivnit pomocí parametrů, které jsou shrnuty v tabulce 3 [7].

6.3.1 Tvorba RRD databáze

Výpis 7 uvádí příklad pro vytvoření RRD databáze uložené v souboru `database.rrd`. V definici databáze jsou nadefinovány proměnné `down` a `up` typu `GAUGE`. Hodnoty 600 jsou maximální prodlevy mezi dvěma záznamy. Následující hodnoty nastaveny na `U` představují minimum nebo maximum velikosti proměnných ve kterých se tyto proměnné v databázi pohybují. Není ale nutné je definovat, proto jsou nastaveny na `undefined` [7].

```
rrdtool create database.rrd --step 300 \
    DS:down:GAUGE:600:U:U \
    DS:up:GAUGE:600:U:U \
    RRA:AVERAGE:0.5:1:600
```

Výpis 7: Vytvoření RRD databáze

6.3.2 Občerstvování RRD databáze

Občerstvování databáze probíhá pomocí příkazu, který je popsán ve výpisu 8. Je potřeba uvést cestu k souboru obsahující RRD databázi, do kterého se následně vloží časová značka a posléze číselné hodnoty proměnných ve stejném pořadí, jak byly definovány ve výpisu 7. Časová značka je vyjádřena v takzvaném UNIX čase [17]. Časová značka a hodnoty veličin se od sebe navzájem oddělují dvojtečkou.

Poznámka 6.3 UNIX čas je celočíselné číslo zobrazující počet vteřin, které uběhly od data 1.1. 1970 [17].

```
rrdtool update database.rrd time:hodnota_down:hodnota_up
```

Výpis 8: Občerstvení RRD databáze

6.4 Vytvoření grafu

Výpis 9 obsahuje příkaz, pomocí kterého `rrdtool` vytvoří graf. Data potřebná pro vykreslení grafu se načtou ze souboru, který plní úlohu RRD databáze.

Poznámka 6.4 Vytvoření grafu lze ovlivnit různými parametry. Níže v textu jsou uvedeny a popsány pouze některé. Více se o generování grafu lze dozvědět na oficiálních stránkách projektu [19]. Informace lze získat také v seriálu Milana Gigela, který popisuje nastavení RRDtool [7].

Parametr	Popis
<code>--title</code>	Nastavuje titulek grafu.
<code>--imgformat</code>	Definuje výstupní typ obrázku [png—gif].
<code>--width</code>	Šířka výsledného grafu v pixelech.
<code>--height</code>	Výška výsledného grafu v pixelech.

Tabulka 4: Základní parametry tvorby grafu

Prvním parametrem definujeme cestu a jméno výstupního souboru do něhož se graf vykreslí. Dalšími parametry ovlivňujeme vzhled výsledného grafu. Lze nastavit nadpis grafu, upravit velikost výsledného obrázku, měnit barvy a podobně.

6.4.1 Graf za libovolné období

Jedním z nejsilnějších argumentů, proč použít aplikaci RRDtool, se skrývá v parametrech `--start` a `--end`. Těmito parametry lze vygenerovat graf za libovolné⁴ období.

Poznámka 6.5 Při definici *start_time* a *end_time* se čas zapisuje ve tvaru UNIX času .

```
rrdtool graph graf.png \
  --title "Jmeno_grafu" \
  --start start_time \
  --end end_time \
  --imgformat PNG \
  --width=800 \
  --height=600 \
  DEF:down=databaze.rrd:down:AVERAGE \
  DEF:up=databaze.rrd:up:AVERAGE \
  AREA:down#00ff00:Download \
  LINE3:up#ff0000:Upload
```

Výpis 9: Vytvoření grafu

⁴Je zapotřebí mít data za vybrané období

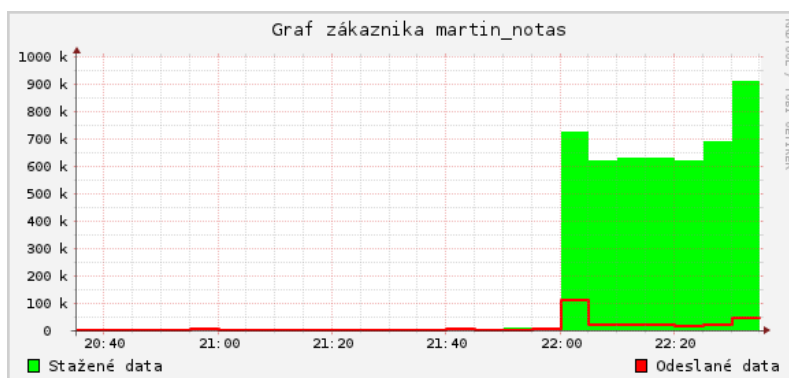
6.5 Návrh aplikace ke tvorbě grafu

Návrh aplikace byl postavené na výsledcích, kterých bylo dosaženo v bakalářské práci *Absolvování individuální odborné praxe* [21]. Aplikace RRDtool bude nainstalována na linuxovém stroji. Ovládání programu RRDtool bude zajišťovat jednoduchý skript, který bude propojen s databází. Z databáze si skript načte údaje potřebné ke svému připojení ke směrovači. Úkoly tohoto skriptu jsou:

- Získat data ze síťového prvku.
- Získané data rozčlenit podle síťových veličin.
- Síťové veličiny zakreslit do grafu.
- Vytvořený graf uložit na pevný disk, resp. občerstvit už vytvořený graf novými daty.

6.5.1 Sledování dat přenesených ke klientovi

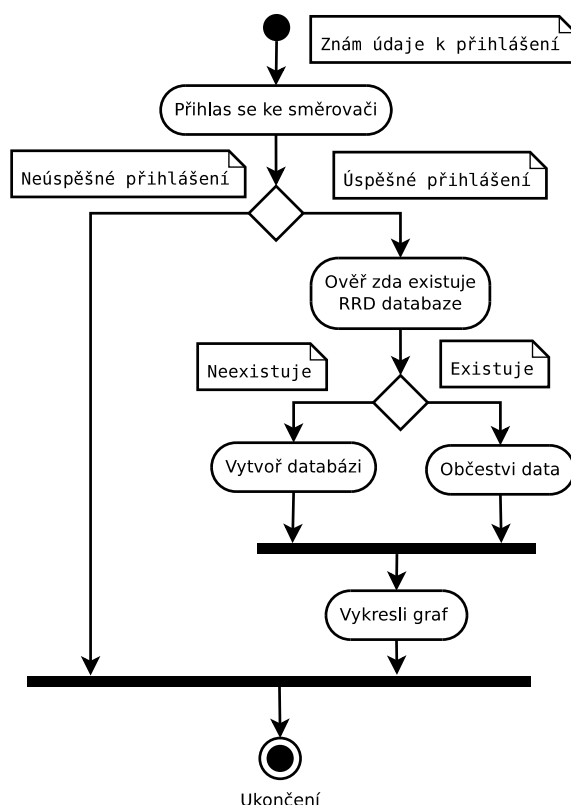
V tomto případě se jedná o aktuální využití přiděleného pásma zákazníkem. Data se získávají na směrovači ze sekce `queue simple`. Tato sekce nabízí k načtení aktuální hodnoty využívané klientem. Sekce nabízí informace o využití pásma směrem k a od klienta. Získané hodnoty se následně zapíší do RRD databáze.



Obrázek 16: Výsledný graf vytížení linky

6.6 Implementace skriptu

Skript je napsán ve skriptovacím jazyce PHP 5. Skript nemá žádný výstup, pouze provádí spuštění aplikace RRDtool. Jediná aplikační logika skriptu spočívá v ověřování existence RRD databáze. Skript je spouštěn v časových intervalech a popis jeho chování je popsán diagramem aktivit na obrázku 17. Skript využívá k ovládání programu RRDtool vestavěnou funkci jazyka PHP `shell_exec()`. Tato funkce přijímá parametr datové typu `string`. Přes tento parametr se zadávají příkazy, které vykonává interpret příkazové řádky [6]. Návrátovou hodnotu představuje řetězec znaků přímo z interpreta.



Obrázek 17: Diagram aktivit popisujících chování PHP skriptu

6.6.1 Spojení se systémem

Skript je závislý na vytvořených pravidlech na směrovači v sekci `queue simple`. Tato pravidla se vytvářejí pomocí rozhraní pro úpravu rychlosti linky. V případě, že směrovač neobsahuje žádné pravidlo, skript nevygeneruje žádnou RRD databázi ani graf. Implementace rozhraní je popsána v kapitole 5. Rozhraní pro úpravu rychlosti linky lze Ovládat pomocí webového informačního systému, jež je popisován v kapitole 8. Pomocí webového rozhraní lze vytvořit i odstranit pravidla na směrovači.

6.6.2 Periodické spouštění skriptu

Aby skript vytvářel smysluplné grafy je zapotřebí jej periodicky spouštět. Skript běží na platformě Linux, proto bylo vhodným řešením použití daemona `cron`. Cron se stará o periodické provádění příkazů zapsaných v konfiguračním souboru [17]. Konfigurační příkaz obsahuje časy spuštění, uživatele který příkaz spustí a příkaz, který se má provést.

6.6.3 Popis konfiguračního souboru cron

Konfigurační soubor se jmenuje `crontable` a nachází se ve složce `/etc/` [17]. V prvních pěti sloupcích se nastavuje čas, ve kterém se skript spouští. Časové sloupce se oddělují mezerou. Pořadí sloupců je následující: [minuta] [hodina] [den v měsíci] [měsíc] [den v týdnu] [uživatel] [příkaz]. Ukázka konfigurace ve výpisu 10 nastavuje spuštění skriptu každých pět minut, každou hodinu, měsíc, den v týdnu. Spuštění skriptu proběhne pod uživatelem *martin*.

Poznámka 6.6 Čas se definuje číslicemi oddělenými čárkou. Zástupný znak hvězdička naznačuje ze se vyhoví ve všech případech. Mezi jednotlivými časovými údaji je rozestup tvořen mezerou. Mezi časovým údajem den v týdnu a jménem uživatele je rozestup tvořený tabulátorem. Tabulátorem je tvořen rozestup i mezi jménem uživatele a příkazem.

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * martin  php ~/Dokumenty/php/create_graph.php
```

Výpis 10: Konfigurace periodického spouštění skriptu

6.7 Souhrn použitého software a jeho verze

Soupis programu použitý při implementaci komponenty na tvorbu grafu je shrnut v tabulce 5.

Název programu	Použitá verze
RRDtool	1.4.3
Ubuntu Linux 64 edition	11.10
Linux kernel	3.0.0.17-generic
PHP	5.3.6-13ubuntu3.6

Tabulka 5: Typ a verze použitých programů

7 Monitorování síťových komponent

Při monitorování síťových komponent se zjišťuje stav směrovače a stavy, ve kterých se nachází síťové rozhraní dostupné na směrovači.

7.1 Stav směrovače

Z pohledu stavu směrovače je důležité sledovat využití dostupných zdrojů, které jsou k dispozici na směrovači. Především je vhodné sledovat stav těchto veličin:

- Velikost dostupné paměti RAM nebo HDD.
- Využití procesorů.

V případě nedostatku disponibilních zdrojů směrovače, je možné naplánovat vylepšení zdrojů na směrovači nebo jeho výměnu za typ s většími parametry.

7.2 Stav síťových rozhraní

Sledování veličin na síťových rozhraních lze rozdělit na dvě základní kategorie. Tou první jsou rozhraní kabelových rozvodů. Do druhé kategorie lze zařadit rozhraní, která jsou bezdrátová.

7.2.1 Drátová síťová rozhraní

U drátových síťových rozhraní jsou důležité tyto veličiny:

- Stav rozhraní: UP, DOWN
- Rychlost komunikace: 10, 100, 1 000 Mbps.
- Typ spojení: half, full duplex.

Zobrazené veličiny prozradí první možné problémy na spojení. Na diagnostikované problémy je třeba soustředit veškerou pozornost.

7.2.2 Bezdrátové síťové rozhraní

U bezdrátových síťových rozhraní je třeba se zaměřit na sledování více veličin než u drátového typu spojení. Na bezdrátové části počítačové sítě se častěji vyskytují výpadky nebo omezení služeb. Zde jsou zpravidla sledovány tyto veličiny:

- Rychlost spojení.
- Hodnota za jak dlouho připojené stanice odpovídají ACK.
- Kvalita připojeného klienta.
- Odstup použitelných signálů od šumu.

Veličiny bezdrátových síťových rozhraní se často mění v čase. Je to dáno různými ovlivňujícími faktory. Faktory ovlivňující kvalitu bezdrátového spojení mohou být:

1. Přírodního charakteru

- Déšť
- Vítr
- Mlha

2. Mechanické

- Poškozený kabel, který vede signál.
- Nedostatečná nebo oslabená izolace vedení signálů.

Vlivem těchto faktorů je vhodné sledovat vývoj veličiny u bezdrátových síťových rozhraní v časových intervalech. Za časový interval je obvykle považován jeden den. K zachycení a zobrazení těchto veličin se nejčastěji používají grafy.

7.3 Návrh implementace monitorování síťových komponent

V podkapitole 7.2.1 a 7.2.2 jsou uvedeny veličiny vhodné ke sledování. Komponenta ke sledování těchto veličin však potřebuje získat data od směrovače. K získání dat se využije již navržená komponenta, která má na starosti komunikaci se směrovači viz kapitola 4.

7.3.1 Implementace monitorování síťových komponent

Ve webové službě byly implementovány třídy `MkSystem` a `MkInterface`, které obsahují funkce k získávání systémových prostředků či detailů k síťovým rozhraním.

7.3.2 Načtení informací o systémových zdrojích

K načtení informací o systémových zdrojích je nutno vyslat požadavek prostřednictvím `RouterController`. `RouterController` odešle požadavek, k získání systémových zdrojů z konkrétního směrovače, webové službě.

Webová služba, reprezentována třídou `RouterDb`, přijímající požadavky od `RouterController`, provede přihlášení k požadovanému směrovači a zavolá funkci `GetResource()` ve třídě `MkSystem`. Funkce provede načtení systémových zdrojů ze směrovače. Získaná data převede do objektové podoby datového typu `SystemResource`.

Datový typ `SystemResource` je návratová hodnota zaslána k prezentaci do webového rozhraní. Webové rozhraní prezentuje získaná data v podobě tabulky se dvěma sloupci. Ukázka prezentace dat je zobrazená na obrázku 18.

Výše popsany postup je zobrazen na obrázku 8 na stránce 27 a zachycuje sekvenční diagram.

Uptime:	2w6d07:52:53
Version:	5.4
FreeMemory:	13792
TotalMemory:	29708
Cpu:	MIPS 24Kc V7.4
CpuCount:	1
CpuFrequency:	400
CpuLoad:	3
FreeMemory:	13792
FreeHddSpace:	32100
TotalHddSpace:	61440
WriteSectSinceReboot:	1799171
WriteSectTotal:	1799171
ArchitectureName:	mipsbe
BoardName:	RB750

Obrázek 18: Prezentace systémových zdrojů

7.3.3 Načtení informací o síťových rozhraních

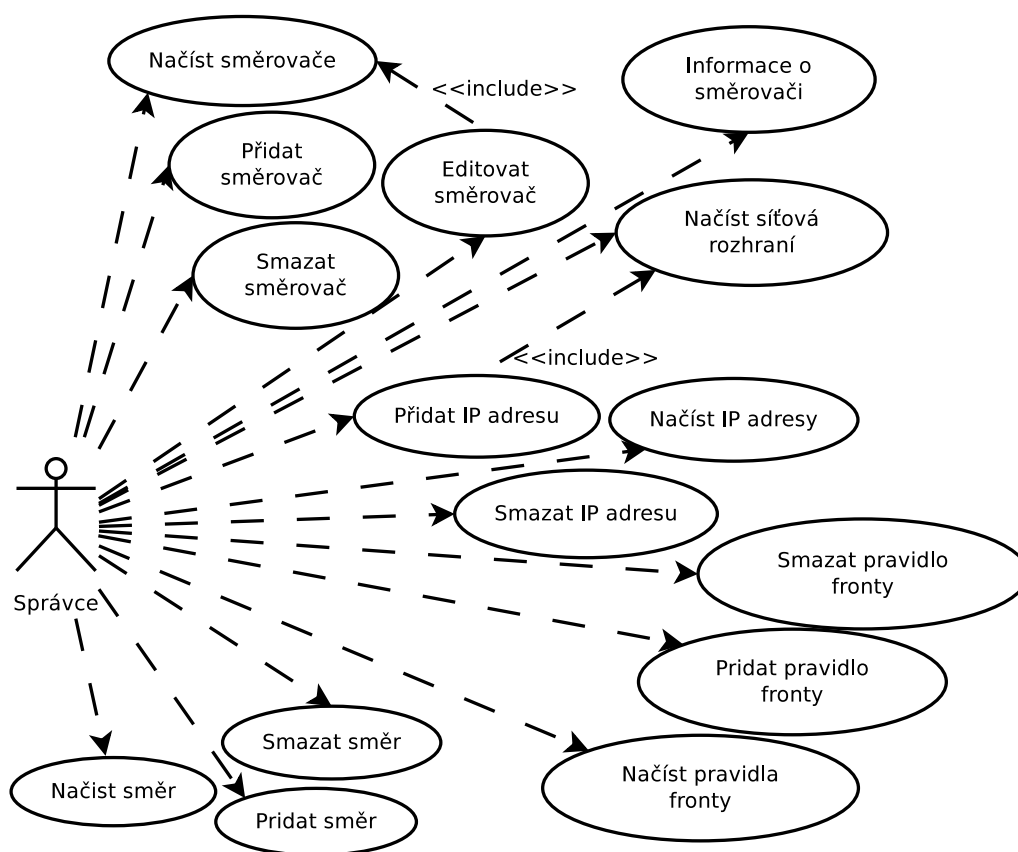
K získání informací o síťových rozhraních je nutno vyslat požadavek prostřednictvím `RouterController.RouterController` zavolá funkci `GetEthernetInterface()` jako požadavek na získání informací o síťových rozhraních. Funkce `GetEthernetInterface()` zavolá funkci `GetAllEthernet()`. Funkce `GetAllEthernet()` se postará o získání informací o síťových rozhraních. Získané informace převede do objektové podoby datového typu `InterfaceEthernet`. Objekty třídy `InterfaceEthernet` jsou vloženy do kolekce, která je předána k prezentaci do webového rozhraní.

8 Návrh webového rozhraní pro správu systému

Webové rozhraní bude sloužit jako vstupně výstupní ovládací rozhraní mezi správcem počítačové sítě a webovou službou. Webové rozhraní musí umět navázat spojení s webovou službou. Komunikace s webovou službou bude zajištěna pomocí vzdáleného volání funkcí, které má webová služba k dispozici. Tyto funkce budou nabízet možnost načíst data ze síťového prvku a také do něj poslat příkaz. Webové rozhraní bude prezentovat funkčnost webové služby.

8.1 Využití webového rozhraní

Webové rozhraní bude nabízet správci přehled o stavu síťových prvků a možnost jeho aktivního ovlivňování. Práce s webovým rozhraním je zachycena na obrázku 19 pomocí diagramu případu užití.



Obrázek 19: Diagram případu užití webového rozhraní

8.2 Vstupy do systému

Vstupy do systému pojednávají o možných vstupních datech, která slouží k evidenci údajů. Údaje, IP adresa, Cesta a Fronta, se evidují ve formě pravidel na směrovačích. Síťový prvek se eviduje ve formě záznamu v databázi.

Síťový prvek (Jméno, IP adresa, uživatelské jméno, heslo)

IP adresa (Síťové rozhraní, IP adresa/rozsah)

Cesta (Cílová IP adresa/maska, IP adresa brány, typ, rozsah, cílový rozsah)

Fronta (Popis, IP adresa, rychlost příchozí, rychlost odchozí)

8.3 Výstupy ze systému

Výstupy ze systému umožní správci náhled pravidel, která jsou evidována na směrovači a záznamech evidovaných v databázi. Výstupy jsou přehledně znázorněny formou webových tabulek.

Seznam síťových prvků (Jméno, IP adresa, uživatelské jméno, heslo)

Informace o směrovači (Čas od spuštění, verze OS, volná paměť RAM, celková paměť RAM, typ procesoru, počet procesorů, frekvence procesorů, vytížení procesorů, volná paměť HDD, celková paměť HDD, počet zapsaných sektorů od startu systému, počet zapsaných sektorů celkem, jméno architektury procesoru, jméno architektury)

Síťové rozhraní (identifikátor, název, MTU, Mac adresa, plný duplex, rychlost, funkčnost)

Seznam IP adresy (identifikátor, IP adresa/maska, adresa sítě, rozhraní, jde o dynamickou adresu)

Fronta (Popis, cílová IP adresa/maska, priorita, typ fronty, omezení, maximální rychlost)

8.4 Nefunkční požadavky

Webové rozhraní bude vytvořeno v prostředí ASP.NET. Aplikační logiku webové aplikace zajistí jazyk C#. Výstup aplikace bude dostupný pomocí webového prohlížeče. Výstup na prohlížeči bude zajištěn značkovacím jazykem (X)HTML. Výstup bude odpovídat W3C standardům webové aplikace.

8.5 Datová analýza

Data, které jsou důležitá pro potřeby funkce systému, se nacházejí na směrovačích. Potřebná data se načítají přímo z konkrétního směrovače. Z pohledu systému řízení báze dat je potřeba uchovávat pouze seznam dostupných směrovačů. Datová analýza je omezená pouze na jednu tabulku, která je popsána datovým slovníkem v tabulce 6.

Jméno	Datový typ	Velikost	Null	Index	Klíč	Integritní omezení	Poznámka
id	integer	-	-	-	PK	-	-
name	varchar	100	ne	-	-	-	-
ip	varchar	100	ne	-	-	-	-
username	varchar	100	ne	-	-	-	-
heslo	varchar	100	ne	-	-	-	-

Tabulka 6: Datový slovník tabulky mikrotik

Název programu	Použitá verze
Windows XP Professional	Verze 2002 SP3
Visual studio 2010	10.0.30319.1
ASP.NET MVC 3 Tools Update	-
SQL Server Compact	4.0
MS .NET Framework	4.0.30319

Tabulka 7: Typ a verze použitých programů

8.6 Implementace webového rozhraní

K implementaci webového rozhraní a k ovládní webové služby bylo zvoleno prostředí ASP.NET. Aplikace byla vybudována na základě návrhového vzoru Model-View-Controller [14], dále jen MVC.

Poznámka 8.1 Prostředí k implementaci nad návrhovým vzorem MVC se musí doinstalovat do prostředí Microsoft Visual Studio⁵.

Implementace controleru a modelu je v jazyce C#. K implementaci pohledu byla použita syntaxe Razor [9], která se liší od klasického ASP.NET přístupu tím, že se data vepisují přímo do HTML prvků. Primárně se nepoužívají ASP komponenty.

Práce s databází byla použita pouze v případě implementace *Nástroje pro zobrazení a správu topologie sítě*, viz kapitola 9.

V ostatních případech controler komunikuje přímo s webovou službou a jako model by se dala považovat právě ona webová služba která zajišťuje práci se směrovači.

⁵Bylo použito MS Visual Studio 2010

8.7 Použitý software k implementaci

V tabulce 7 je uveden seznam použitých softwarových produktů na kterých byla implementace vyvíjena a byla otestována její funkčnost. Při nasazení aplikace do produkčního prostředí však může dojít k nekompatibilitě použitého softwaru.

9 Nástroj pro zobrazení a správu topologie sítě

Nástroj pro zobrazování a správu topologie má dva úkoly:

1. Evidovat síťové prvky, které se budou v rámci webového rozhraní sledovat.
2. Nastavení a změnu toku dat v síti.

9.1 Návrh evidence síťových prvků

Webové rozhraní bude nabízet možnost vkládat do evidence nové síťové prvky, upravovat evidované záznamy, zobrazovat přehled o evidovaných síťových prvcích s možností odstranění nepotřebných síťových prvků.

Evidence síťového prvku obsahuje údaje potřebné k navázání spojení se směrovačem a k identifikaci evidované položky. Evidence bude vedena v databázi. Grafický pohled na strukturu databáze nabízí tabulka 6 na straně 47, která zobrazuje datový slovník.

9.1.1 Vložení nového síťového prvku

K vložení nového síťového prvku bude sloužit webový formulář. Webový formulář bude na samostatné stránce. Přístup na tuto stránku bude zajišťovat tlačítko umístěné na stránce s přehledem síťových prvků. Webový formulář k vložení nového záznamu bude požadovat vyplnění těchto údajů:

- Textový popis síťového prvku sloužícího k identifikaci.
- Cílovou IP adresu síťového prvku.
- Uživatelské jméno a heslo potřebné k připojení.

9.1.2 Přehled evidovaných síťových prvků

Přehled evidovaných síťových prvků bude zobrazen na vlastní webové stránce a bude zajištěn webovou tabulkou. Webová tabulka bude obsahovat evidované záznamy, přičemž každý záznam bude na vlastním řádku. První řádek tabulky bude obsahovat názvy sloupců. V posledním sloupci budou nabídnuty možnosti pracovat s konkrétním záznamem.

9.1.3 Práce se záznamem

Práce se záznamem budou zajišťovat tři tlačítka, která zajistí požadovanou funkčnost.

Detail Tlačítko detail umožní zobrazit detailní informace o záznamu. Detail se zobrazí na samostatné stránce.

Mazání Tlačítko smazání umožní odstranit záznam z evidence.

Úprava záznamu Tlačítko úprava záznamu zajistí přechod na samotnou stránku, na které bude vyplněný formulář s možností úpravy záznamu.

9.2 Návrh nastavení a změna toku dat v síti

Změnu toku dat v síti lze ovlivnit směrovací tabulkou na směrovači. Se změnou toku dat v počítačové síti úzce souvisí změna IP adres na síťových rozhraních. Schopnost měnit IP adresu na síťových rozhraních je důležitá z důvodu dostupnosti sítí do kterých se směřuje datový tok.

9.2.1 Návrh funkcí na změnu toku dat v síti

Webová služba bude mít implementované funkce pro načtení seznamu IP adres a cest do dostupných sítí. Dále budou mít implementovány operace pro vložení nových cest a IP adres, odstranění nepotřebných IP adres a cest na směrovači.

9.2.2 Návrh webového rozhraní pro změnu toku dat v síti

Webové rozhraní využije funkce webové služby k načtení a nastavení IP adres a cest do počítačových sítí. Výčet seznamu cest a IP adres, které jsou dostupné na směrovači, nabídne webové rozhraní formou tabulky. Tabulky budou zvlášť pro každou kategorii. Vložení nové cesty a IP adresy bude umožněno formou webového formuláře. Webový formulář bude také pro každou kategorii zvlášť. Funkce k vložení a odstranění cest na směrovači budou nabídnuty přes funkční tlačítka webového rozhraní. K vložení IP adresy je potřeba, aby uživatel zadal vhodnou IP adresu včetně masky pod sítě a příslušné síťové rozhraní, na které bude IP adresa přidělena.

10 Závěr

V rámci diplomové práce byla vytvořena webová aplikace schopná přihlásit se a pracovat se směrovači platformy Mikrotik. Implementována a odzkoušená byla obousměrná komunikace aplikace směrovače. K ověření funkčnosti byl vytvořen jednoduchý informační systém pomocí něhož se mohl směrovač ovládat. Zadání diplomové práce bylo splněno ve všech bodech až na částečné nesplnění bodu čtyři. V tomto bodě byl záměr vytvořit síť topologie směrovačů, která by poskytla přehled o stavu a rozvětvení sítě. K dokončení záměru vytvořit síť topologie směrovačů byla přislíbená spolupráce s firmou zabývající se poskytováním internetových služeb. Spolupráce s firmou měla spočívat ve zpřístupnění jejich topologie za účelem vytvoření grafu sítě směrovačů v informačním systému. Spolupráce byla neustále odkládána až se nakonec neuskutečnila vůbec. Z tohoto důvodu byl záměr vytvořit graf sítě omezen pouze na evidenci směrovačů.

Při tvorbě praktické části diplomové práce jsem si vyzkoušel komunikaci prostřednictvím proprietárního rozhraní Mikrotik API a tvorbu rozhraní pro správu směrovače. Také jsem měl možnost použít výsledky své bakalářské práce a obohatit tím monitorovací funkce implementované webové aplikace. Při tvorbě práce jsem použil jak vědomosti získané v průběhu studia tak i nové přístupy k tvorbě programu. Například pro vytvoření informačního systému jsem využil návrhového vzoru model view controller v prostředí ASP.NET. V době kdy jsem studoval předměty zabývající se tvorbou aplikací v prostředí ASP.NET, byl MVC způsob vývoje aplikací ve fázi přípravy. Informace o MVC byly pouze v teoretické úrovni a jsem velice rád, že se mi povedlo v tomto návrhovém vzoru vytvořit ovládací rozhraní webové služby. Jsem přesvědčen o tom, že tvorba aplikace s pomocí návrhového vzoru MVC, dostává větší logiku a hlavně zpřehlední aplikaci.

Aplikace je připravena k využití na počítačové síti, která je postavena na platformě Mikrotik. Vzhledem k rozdělení aplikace na část komunikující se směrovačem a část starající se o obsluhu komunikační části, lze využít pouze hlavní část určenou ke komunikaci a začlenit aplikaci do stávajícího informačního systému firmy.

Do budoucna je možno aplikaci dále rozšířit o možnost ovládání dalších funkcí směrovače. Další eventualitou, jak rozšířit využití aplikace, je možnost komunikovat se směrovači postavenými na jiných platformách.

11 Reference

- [1] BOUŠKA, Petr. LUGSCH. Zbyszek. PUŽMANOVÁ. Rita. Sítové technologie přehledně a stručně. *Časopis Connect* 4 2010 ročník XV s.8-20.ISSN 1211-3085
- [2] Český překlad Obecné veřejné licence GNU, *GNU General Public License*
<http://www.gnugpl.cz/> [cit. 31.3.2012]
- [3] DEVAINE Max, 2008 *Nagios + Centreon + MySQL - instalace a základní konfigurace*
<http://www.abclinuxu.cz/clanky/site/nagios-plus-centreon-plus-mysql-instalace-a-zakladni-konfigurace> [cit. 31.3.2012]
- [4] DOSEDĚL, Tomáš. Nabídka operátorů pro firemní zákazníky. *Časopis Connect* 9 2009 ročník XIV s.8-10.ISSN 1211-3085
- [5] FROOM, Richard. SIVASUBRAMANIAN, Balaji. FRAHIM, Erum. *Implementing Cisco Switched*, Indianapolis, IN 46240 USA, Cisco Press, 2010
- [6] Function `shell_exec`, *PHP:shell_exec - Manual*
<http://php.net/manual/en/function.shell-exec.php> [cit. 17.4. 2012]
- [7] GIGEL, Milan. 2001 *Prechádzame na RRDTOOL*
<http://www.root.cz/serialy/prechadzame-na-rrdtool/> [cit. 3.4.2012]
- [8] GRYGÁREK, Petr. *Referenční model ISO OSI*
http://www.cs.vsb.cz/grygarek/PS/lect/OSI_RM.html [cit. 12.4.2012]
- [9] Intro to ASP.NET Web Programming Razor Syntax : Official Microsoft Site *Introduction to ASP.NET Web Programming Using the Razor Syntax*.
<http://www.asp.net/web-pages/tutorials/basics/2-introduction-to-asp-net-web-programming-using-the-razor-syntax> [cit. 1.5.2012]
- [10] ISPadmin, *ISP admin Wiki*
http://wiki.ispadmin.eu/index.php/Main_Page [cit. 16.4.2012]
- [11] KALÁB, Dalibor. IP telefonie znovu a silnější. *Časopis Connect* 5 2010 ročník XV s.8-9.ISSN 1211-3085
- [12] Mikrotik team, *Manual:API in C Sharp*
http://wiki.mikrotik.com/wiki/API_in_C_Sharp [cit. 12.4.2012]
- [13] Mikrotik team, *Manual:API*
<http://wiki.mikrotik.com/wiki/Manual:API> [cit. 31.3.2012]
- [14] MVC : Official Microsoft Site *Getting Started with ASP.NET MVC 3*
<http://www.asp.net/mvc> [cit. 1.5.2012]
- [15] NAGEL, Christian EVJEN Bill, GLYN Jay, C# 2005 *Programujeme profesionálně*, Brno: Computer Press, 2006

- [16] Nagios *The Industry Standard in IT Infrastructure Monitoring*
<http://www.nagios.org/> [cit. 31.3.2012]
- [17] NEMETH, Evi. SNYDER, Garth. HEIN R. Trent. *Linux - kompletní příručka administrátora*, Brno: Computer Press, 2004
- [18] ODVÁRKA, Petr. Svět sítí, 2000 *Ethernet*
<http://www.svetsiti.cz/clanek.asp?cid=Ethernet-1992000> [cit. 12.4.2012]
- [19] OETIKER, Tobias. *About RRDtool*
<http://oss.oetiker.ch/rrdtool/index.en.html> [cit. 3.4.2012]
- [20] PASEKA, Radim. *Omezování nežádoucího provozu na bezdrátových sítích*, VŠB-TU 2004
http://www.cs.vsb.cz/grygarek/SPS/materialy/Paseka_DP/dp.pdf
- [21] PATYK, Martin. *Absolvování individuální odborné praxe*, VŠB-TU 2010 [cit. 17.4.2012]
- [22] Router OS *MikroTik Routers and Wireless*
<http://www.mikrotik.com/software.html> [cit. 16.4.2012]
- [23] The Dude *MikroTik Routers and Wireless*
<http://www.mikrotik.com/thedude.php> [cit. 31.3.2012]
- [24] Ústavní soud ČR, 2011, *Ústavní soud zrušil část zákona o elektronických komunikacích*
<http://www.concourt.cz/clanek/5068> [cit. 16.4.2012]
- [25] ZANDL, Patrick. *Bezdrátové sítě WiFi Praktický průvodce*, Brno: Computer Press, 2003